LEARNING CONTROL OF INVERTED PENDULUM SYSTEM BY NEURAL NETWORK DRIVEN FUZZY REASONING

- The Learning Function of NN-Driven Fuzzy Reasoning under Changes of Reasoning Environment -

Isao HAYASHI, Hiroyoshi NOMURA and Noboru WAKAMI

Central Research Laboratories, Matsushita Electric industrial Co. Ltd.
3-15, Yagumo-nakamachi, Moriguchi, Osaka, 570 Japan

Whereas conventional fuzzy reasonings are associated with tuning problems which are lack of membership functions and inference rule designs, a neural network driven fuzzy reasoning (NDF) capable of determining membership functions by neural network is formulated. In the antecedent parts of the neural network driven fuzzy reasoning, the optimum membership function is determined by a neural network, while in the consequent parts, an amount of control for each rule is determined by another plural neural networks. By introducing an algorithm of neural network driven fuzzy reasoning, inference rules for making a pendulum stand up from its lowest suspended point are determined for verifying the usefulness of the algorithm.

1. INTRODUCTION

Extensive applications of fuzzy reasoning for various control problems, and a number of actual examples of fuzzy control are reported [1] lately. However, the fuzzy reasoning is generally involved with a tuning problem [2], that is, the form of fuzzy number, and the fuzzy variables of antecedent parts and consequent parts of fuzzy inference rules, have to be adjusted for minimizing the difference between the estimation of fuzzy reasoning and the output data for given input data.

As a method to solve the tuning problem, a neural network driven fuzzy reasoning (NDF) [3, 4] by which inference rules are constructed from the learning function of neural network [5, 6] is previously reported. The NDF is a type of fuzzy reasoning having an error back - propagation type network [7] which represent fuzzy sets in its antecedent parts, while another error back - propagation type network represents an input - output relationship between input and output data of consequent parts of each rule.

In this paper, an algorithm for constructing inference rules based on NDF is introduced first, and an experimental verification of its effectiveness is performed taking an example for an inverted pendulum system.

In this experiment, a pendulum in its hanged position is surely swang up and is held at an inverted position by using a mechanism controlled by inference rules which are constructed by determining fuzzy sets from the observations of pendulum operator by utilizing NDF algorithm.

The inference period required for controlling the swing-up motion of pendulum is approximately 15 msec. As a parameter which governs the dynamic characteristics of inverted pendulum system, the length of pendulum is considered here, and changes of control characteristics of NDF caused by this is studied. Since the fuzzy set of antecedent parts and input - output relationship of consequent parts can be determined by means of NDF without finetuning of inference rules by utilizing the learning function of neural network acquired from the input-output data, it is an advantageous

method to solve tuning problems of fuzzy reasoning.

2.. ARTIFICIAL NEURAL NETWORK DRIVEN FUZZY REASONING (NDF)

The NN – driven fuzzy reasoning (NDF) is a fuzzy reasoning [8] using linear functions in its consequent parts. In a NDF, the membership functions in the antecedent parts is determined in a multi–dimensional space. For example, the following rules R1, R2, and R3 of the conventional fuzzy reasoning wherein x1 and x2 are input variables, y1, y2, and y3 are output variables, and a10 and a11 are coefficients, and FSL and FBG are fuzzy numbers where SL and BG mean small and big respectively, are considered.

R1 ; IF x1 is FSL and x2 is FSL,
THEN y1 = a10 + a11x11 + a12x12
R2 ; IF x1 is FSL and x2 is FBG,
THEN y2 = a20 + a21x21 + a22x22 (1)
R3 ; IF x1 is FBG,
THEN y3 = a30 + a31x31

Since the above condition means that x1 is small and x2 is small in the antecedent parts, the fuzzy sets F1 = FSL ∩ FSL can be constructed in a partial space of the input as shown in Fig. 1. The same can be applied for the fuzzy sets to be constructed for R2 and R3 likewise. Since the boundary between the each partial space is vague, the boundary is shown by the hatched lines. That means that the input space consisted of x1 and x2 is divided into individual partial spaces by the number of fuzzy rules, and the fuzzy sets of antecedent part of each inference rules are constructed in each partial space, while the NDF is determined by the fuzzy sets of antecedent parts by utilizing the back – propagation type network.

An explanation for the back – propagation type network is as follows. Since neural networks are constrained by a general type processing unit found in the neural system, and the processing unit in a neural network shares some of the physical properties of real neurons, the processing unit is called neuron here.

Fig. 2 shows an example of fundamental layered back propagation type networks containing M layers, where the first layer is called an input layer, the M–th layers are output layers, and other layers are called intermediate layers. Every neuron within these layers represents respective correlation between the multi–inputs xij and multi–outputs yi expressed by the following equations.

$$yi = f\left(\sum_{j=1} \alpha ij\ xij + \theta\right) \tag{2}$$

$$f(Z) = \frac{1}{1 + \exp(-Z)} \tag{3}$$

where θ is a weight showing a correlation between neurons.

In this paper, for a given input and output expressed by x = (x1, x2, ... , xn) and y = (y1, y2, ... , yw) respectively, the input–output correlation of back–propagation type network as a whole is expressed by;

$$y = NN(x) \tag{4}$$

The structure of model function NN (x) is characterized by M–layers [u1 X u2 X .. X uM] where ui , i = 1, 2, ... , M are the numbers of neurons within the input, hidden and output layers respectively. Fig. 2 shows a structure of back propagation type network consisting of four–layers [3 X 2 X 2 X 2].

Fundamental considerations made on the NDF are that the model equations y1, y2, and y3 in the consequent parts are identified as the non – linear Eq. (4) for obtaining the model equations.

The fundamental consideration made on the membership functions in the antecedent parts is a method shown in Fig.

3.

If the relationships between rules R1, R2, R3 and input data (xi1, xi2) where i = 1, 2, ... , N are considered, the first data x1 are (x11, x12) = (0.2, 0.15), and these data belong to rule R1. Thus the data attribution to the rule can be expressed by (R1, R2, R3) = (1, 0, 0). The back-propagation type network three-layers [2X3X2] of which input and output layer are (X11, x12) and (R1, R2, R3) respectively can be derived from the input-output data utilized in the learning process. However, the maximum number of learning is limited to be less than about 1000.

When another data different from the input-output data are assigned to the neural network, the estimated values of back propagation type neural network are considered as membership values of fuzzy sets in the antecedent parts since the estimated value represents the attribution of data to each rule. A rule division performed by NDF is typified in Fig. 4 which shows non-linear divisions unlike the rectangular divisions shown in Fig. 2.

Pao proposed a method for determining fuzzy sets by using a neural network [9], and obtained intersections and union sets of fuzzy sets. However, what he carried out were the determinations of intersection and union sets of fuzzy sets from the coupling patterns between each unit of neural network, and was not the type determining the shape of fuzzy sets from the input-output data such as excutable by NDF.

In a NDF, the control rules are represented by an IF-THEN format shown below.

Rs ; IF x = (x1, x2, ... , xn) belongs to As,
THEN ys = NNs(x1, x2, ... , xm)
where s = 1, 2, ... , r, m≤n (5)

The number of inference rules employed here is expressed by r, and As represents a fuzzy set in the input space area of antecedent parts. The degree of belongings of input x = (x1, x2, ... , xn) to the s-th inference rule is defined to as the membership value of fuzzy sets As to the input x. Furthermore, the amount of operations ys of consequent parts is an estimated value for a case where a combination of input variables (x1, x2, ... , xm) is substituted in the input layer of back propagation type network, wherein the number of variables employed in this case is m according to a method for selecting the optimum model employing back-propagation type network.

Although it is also possible to determine an overall non-linear relationship by using only one back-propagation type network, the determination of overall input-output relationship by applying back-propagation type network for each partial space is considered more advantageous than employing only one back-propagation type network for better clarification of overall non-linear relationship.

In order to carry out an optimum model selection for the back-propagation type network of consequent parts, a stepwise method [10] by which a specified input variable derived from a combination of input variables is introduced and removed for obtaining a model which outputs an optimum estimated value, is available.

In the present work, only an elimination of input variables from a combination of input variables by utilizing back-propagation type network is performed for deriving an optimum combination of input variables and model formula. A summation of the second powers of residuals is employed for evaluating and dtermining the input variables.

An explanation for the algorithm of NDF is given in the following referring a block diagram of NDF shown in Fig. 5. The stepwise procedures taken for obtaining the inference rules and the control value yi* for the input data xi are as follows.

Step 1: Selection of input variables, x1, x2, ... , xn, which are related to the control value y. This is for an assumed case where the input

– output variables (yi, xi) = (yi, xi1, xi2, ... , xin) where i = 1, 2, ... , N, are obtained and the input data xij where j = 1, 2, ... , n, are the i-th data of input variable xj.

Step 2: Division of input-output data into r classes of Rs where s = 1, 2, ... , r. As mentioned before, each partition is regarded as an inference rule Rs, and the input-output data for each Rs are expressed by (yi(s), xi(s)) where i = 1, 2, ... , Ns providing that Ns is a number of input-output data for each Rs.

Step 3: Decision of membership functions in the antecedent parts by using the neural network NNmem shown Fig. 5 providing that the structure of a back-propagation type network is a M-layered [n × u2 × ... × uM-1 × r]. The method for determining the form of membership functions is described previously.

Step 4: Decision of control models in the consequent parts by using the neural networks NN1, NN2, ... , NNr shown in Fig. 5 providing that the structure of each back-propagation type network NNs is a M-layered [k × u2 × ... × uM-1 × 1] where k = n, n-1, ... , 1, and selections of optimum model for each NNs are performed.

Consequently, the stepwise procedures for determining input variables by utilizing back-propagation type network, and the method for determining the structure of consequent parts are described in the following.

Setting a condition at k = n, the input variables xi = (xi1, xi2, ... , xik) where i = 1, 2, ... , N, are assigned for the input layer of each NNs, and the output variables yi is assigned for the output layer of each NNs, where the input variables assigned for the input layer and the output variables assigned for the output layer are respectively expressed by:

$$s = \{x1, x2, \ldots, xk\} \quad (6)$$

$$s = \{y\} \quad (7)$$

where s represents a set of input variables assigned for the input layer of each back-propagation type network NNs, and s represents a set of output variables assigned for the output layer of NNs.

An estimation eyi for the input data xi1, xi2, ... , xik can be obtained after repeated learnings made on the back-propagation type network of NNs. However, the number of learnings is set at approximately 3000. Then the sum of mean squared errors of the output data yi and estimation eyi is calculated for obtaining an evaluation value ⊛ks required for determining the input variables.

$$\circledast ks = \left(\sum_{i=1}^{N}(yi - eyi)^2\right)/N, \quad s = 1, 2, \ldots, r. \quad (8)$$

In order to study the degree of correlation of the input variables xj to the output variables y, the input variable xj is temporarily removed from the set of input variables {x1, x2, ... , xk}. The input data from which the input variables xj is removed, xi1,... xij-1,... xij+1, ... , xjk where i = 1, 2, ... , N, are assigned to the input layer of M-layer of the back propagation type network [k-1 u2 ... uM-1 1], and the output data yi are assigned to the output layer. Then, the estimation eyi' for the input data xi1, ... , xij-1, ... , xij+1, can be obtained after the back-propagation learning. An evaluation value ⊛k-1sj required for determining the input variables is derived by calculating the sum of mean squared errors of the output data yi for this estimation eyi'.

$$\circledast k\text{-}1sj = \left(\sum_{i=1}^{N}(yi - eyi')^2\right)/N, \quad s = 1, 2, \ldots, r. \quad (9)$$

The same calculations are conducted for the input variables other than xj for determining the evaluations ⊛k-1s1, ⊛k-1s2, ... , ⊛k-1sj , ... , ⊛k-1sk. The calculation of evaluation which takes a minimum value,

$\Theta k-1sc$, can be obtained by;

$$\Theta k-1sc = \min \Theta k-1sj, \quad \text{where } j = 1, 2, \ldots, k. \tag{10}$$

Eq. 10 shows that the evaluation $\Theta k-1sc$ obtained by removing the input variables xc from the set of input variables takes a minimum value among evaluations $\Theta k-1s1$, $\Theta k-1s2, \ldots, \Theta k-1sj, \ldots, \Theta k-1sk$. By comparing the value of $\Theta k-1sc$ of Eq. 10 to the value of Θks of Eq. 8, the set of variables, Λs, is altered as follows.

$$\Lambda s = \{x1, x2, \ldots, xc-1, xc+1, \ldots, xk\}, \quad \text{If } \Theta k-1sc < \Theta ks \tag{11}$$

$$\Lambda s = \{x1, x2, \ldots, xk\}, \quad \text{If } \Theta k-1sc \geqq \Theta ks \tag{12}$$

When Eq. 11 is established, the sum of mean squared errors can be decreased by removing the input variables xc, and this means that the estimation eyi' represents yi better than eyi.

Therefore, the correlation of input variables xc to the output variables y is considered weak, and the input variables are removed from the input variable sets Λs. As a result of this, a set of newly established input variables is then consisted of k−1 input variables.

On the other hand, the effectiveness obtained by removing input variables temporarily can not be attained when Eq. 12 is established, and this fact means that the input variables xc are strongly correlated with the output variables y, and the number of sets of input variables Λs is left unchanged as k.

In cases where the input variables can be reduced, k is altered to n−1, n−2, ..., 1, and Step 4 is repeated until Eq. 12 can be established, and the procedures for reducing the input variables of back−propagation type network NNs are completed until Eq. 12 can be established.

Thus, the back − propagation type network NNs having the final set of input variables, $\Lambda s = \{x1, x2, \ldots, xm\}$ obtained at the time of procedure completion, becomes an optimum back − propagation type network representing the structure of consequent parts of rule Rs. The same step procedures are conducted for each NNs for determining the consequent parts of all the inference rules. This procedure to reduce the number of input variables is called a stepwise variable reduction method utilizing back − propagation type network.

Step 5: The estimation yi* can be derived by the equation shown below.

$$yi* = \frac{\sum_{s=1}^{r} \mu As\ (xi1, xi2, \ldots, xin) \times meyi(s)}{\sum_{s=1}^{r} \mu As\ (xi1, xi2, \ldots, xin)} \tag{13}$$

$$i=1, 2, \ldots, N.$$

where meyis is an estimation obtained by the optimum back − propagation type network derived by Step 4.

Fig. 5 shows that the estimation yi* can be derived from the results obtained by conducting product operations between the membership values of antecedent parts of each inference rules, or $\mu As(xi1, xi2, \ldots, xin)$ and the estimation of consequent parts, or meyi (s), and by conducting summation operations between each rule continuously. However, Fig. 5 shows a case where a condition of $\mu As(xi1, xi2, \ldots, xin) = 1$ is established.

3. APPLICATION TO INVERTED PENDULUM SYSTEM

The NDF proposed by the authors is capable of forming inference rules automatically, i. e., the function of self − autotuning, and proposed here is an inverted pendulum system to which a learning function by using a NDF is applied. In the algorithm employed for the experiment, four inputs and one output data are acquired by observing manual operating controls, and

fuzzy inference rules and membership functions are then automatically constructed from the acquired data by using the algorithm of NDF.

Fig. 6 shows a structure of inverted pendulum system consisting of four elements explained in the following;

1) Cart which runs on a rail.
2) Pendulum rotatable freely around an axis of cart.
3) Motor which drives the cart.
4) Fixed pulleys and belt system which connect above three parts.

The pendulum angle apart from the perpendicular θ degree and the distance from the original position of cart are detected by the potentiometer b and a shown in Fig. 6 respectively. These are digitized by an AD converter, and the digitized signals are fed to a personal computer wherein the velocities of inverted pendulum angle and the cart distance are calculated from the differences in those obtained at every sampling. The output for the motor control system is then calculated from four variables, i.e., the pendulum angle, angular velocity, cart distance, and the cart velocity by using an algorithm of NDF. As the motor control signal is derived by a personal computer in a digital form, this is converted into an analog value through a DA converter.

The inverted pendulum system has two control areas consisting of a linear-controlling area where the pendulum is standing, and a non-linear controlling area where the pendulum falls. The authors constructed an inverted pendulum system in the linear - controlling area by using a conventional fuzzy control, and a control model constructed in the non - linear controlling area by utilizing NDF, is reported here.

The configuration of inverted pendulum system and the control computer are as follows.

Body : Length of 1,410mm; width of 400mm, height of 880mm.
Pendulum : Length of 400mm , weight of 40g , diameter of 4mm.
Drive force : 25W DC motor with a gear ratio of 12.5 : 1.
Sensors : Potentiometer to measure the distance from the original position of the cart, and another potentiometer to measure the pendulum angle.
Micro-computer : CPU 80286
Program : C-language, 21K bytes.

The preparation of control rules applicable to an inverted pendulum made according to an algorithm developed for constructing the inference rules by applying NDF is now described in the folowing.

Step 1: Preparation of input - output data. This is acquired by an operator who tries to swing up a pendulum by moving the cart right or left direction on the rail by pressing either of corresponding controller buttons until the pendulum is brought to its inverted position, and the following input-output data with a sampling period of 4 msec are recorded:

Output variable
y : Motor control signal (V).
input variables
x1 : Distance from the original cart position.
x2 : Velocity of x1 (cm/sec).
x3 : Pendulum angle (deg).
x4 : Velocity of x3 (deg/sec).

where the input variables x2 and x4 are derived from the differences produced in x1 and x3 values. Approximately 1,000 to 3,000 data are acquired from these manual operations, and from these, 98 input-output data shown in Table 1 applied for the NDF are extracted.

Step 2: Setting of two rules for the input-output data considering data distributions.

Step 3: Determination of membership functions of antecedent parts. A three - layered [4 × 6 × 2] back - propagation type network employed here for determining the antecedent part construction is employed here, and the number of learnings is set at about 1000.

Step 4: Determination of consequent part structure. A three-layered [k×6×1] where k = 4, 3, 2, 1, back-propagation type network for determining the consequent part structure is employed here, and the number of learnings of each back-propagation type network is set at about 3000.

By using a stepwise variable reduction method, we obtain;

$$\circledast 41 = 0.016 \quad (14)$$

$$\circledast 311 = \min\circledast 31j\ (= 0.007),\ j = 1, 2, 3, 4. \quad (15)$$

Therefore,

$$\circledast 311 < \circledast 41 \quad (16)$$

Thus, by removing the input variables x1, we obtain $\Lambda s = \{x2, x3, x4\}$. As for $\Lambda s = \{x2, x3, x4\}$, a stepwise variable reduction method is applied again. By combining Eqs. 8, 9, and 10, we obtain the followings.

$$\circledast 311 = 0.007 \quad (17)$$

$$\circledast 213 = \min\circledast 21j\ (= 0.021),\ j = 2, 3, 4. \quad (18)$$

This means,

$$\circledast 213 > \circledast 311 \quad (19)$$

Thus, no reduction of input variables is made, and the algorithm for Rule 1 is completed by the second calculation process. The inference rules consequently obtained by these are as follows:

R1 ; IF x = (x1, x2, x3, x4) belongs to A1,
THEN y1 = NN1(x2, x3, x4),
R2 ; IF x = (x1, x2, x3, x4) belongs to A2,
THEN y2 = NN2(x1, x2, x4) (20)

Photographs 1 and 2 show the swing-up motions of pendulum controlled by fuzzy inference rules expressed by Eq. (20). Photograph 1 shows sequential motions of pendulum swang from its stable equillibrium state to an inverted stand-still state. The estimation yi* can be derived from Eq. (13). The pendulum can be surely brought to its inverted position regardless the cart position on the rail, or a disturbance applied to the pendulum. Photograph 2 shows the controls of swing-up motion for various given pendulum angles.

An experimental study for the limitation of control performed by NDF is conducted by changing the parameters which govern the dynamic characteristics of controlled object, and the length of pendulum is taken as a parameter governing the dynamic characteristics of pendulum here. The initial position of cart is set at the center position of belt on which the inverted pendulum device is mounted, and the pendulum angle is set at 0 degree when it is hanged down initially and ±180 degree is specified when the pondulum is in an inverted position. The angle is incremented for its clockwise rotation, and decremented for its anti-clockwise rotation.

The inference rules are constructed for a case where pendulum length is 40 cm, and Fig. 7 shows a response of pendulum of such. Figs. 8, 9, and 10 respectively show the responses of the 20, 30, and 50 cm long pendulums. The shifts of pendulum angle are shown by solid lines, and the changes of angular velocity are shown by broken lines in these figures. However, only the changes of pendulum angle and angular velocity until the pendulum comes to an inverted position, and no response after completion of inversion are shown there.

As for the learning of inverted pendulum, the swing-up process of pendulum is learnt for constructing an inference rules applicable to the process of pendulum starting from the hanged down postion to a nearly inverted position. The inverted position is defined as a pendulum angle close to ±180 degree and its angular velocity nearly zero at that time.

As shown in Fig. 7, the pendulum reaches at −180 degrees at 5.4 seconds after starting of control attaining an angular velocity of about 0 deg / sec, and the

pendulum stand still at an inverted position. This is rather natural consequence since an inference rules are established for a 40 cm long pendulum.

In a case where the length of pendulum is set at 20 cm as shown in Fig. 8, a large velocity change is observed, and the angle became 180 degrees at 6.2 seconds attaining an angular velocity of about 0 deg /sec. Although the pendulum reaches at an inverted position and stays there, the angular velocity is larger and a longer lead-in period is required.

Fig. 9 shows a transient response of a 30 cm long pendulum. The pendulum is brought to its inverted position showing a response similar to that obtained with the 40 cm long pendulum, but the angle reaches at −180 degrees at 3.9 sec yielding a higher anlgular velocity which equals to about one half of that obtained with the 20 cm long pendulum. The overall controllable characteristics is silimar to that of 40 cm long pendulum.

Fig. 10 shows a transient response obtained with a 50 cm long pendulum which was unable to brought to its inverted position. As seen in Fig. 10, the pendulum angle could not be brought to its ±180 degree position despite of longer lead-in period. The correlation between dynamic characteristics of pendulum and the variable length of pendulum can be summarized as follows.

1) By applying a NDF to a pendulum system of which length is varied from 40 to 20 cm, a stable operation to bring the pendulum to its inverted position became feasible despite of lead−in period required for its motions. That is to say, the robustness of NDF is higher for the shorter length pendulum.

2) for the cases of longer pendulums, however, the suppression of deviations of the control system can not be attained, and this means that a relearning or additional learning is necessary for the NDF applied for a longer pendulum.

4. CONCLUSION

While the conventional fuzzy reasoning is associated with inherent tuning problems, NDF is, upon input − output variables are given, capable of determing an optimum inference rules and membership functions by utilizing its nonlinearity of back-propagation type network and learning capabilities. In order to verify the usefullness of NDF, it is applied to an experimentally constructed pendulum system wherein the pendulum is brought to its inverted position and stayed there starting from its stable hanged position. The length of pendulum is also altered for confirming itrs effects on the control characteristics of NDF.

Since this method is capable of deriving an inference rules by using the learning function of back-propagation type network, the learning function can be introduced in the fuzzy control. The development of learning function adaptive to the changes of dynamic inference environment should be an important subject to be discussed in future.

5. REFERENCES

1) Hirota, Kaoru, " Robotics and Automation Industrial Applications in Japan", 3rd IFSA Congress, Seattle, 1989, pp.229-230.

2) Lee, Chuen-Chien, "A Self-Learning Rule - Based Controller with Approximate Reasoning", Memorandom, No. UCB / ERL, M89/84, Univ. Calf. Berk, 1989.

3) Hayashi, Isao, Nomura, Hiroyoshi and Wakami, Noboru, "Artificial Neural Network Driven Fuzzy Control and it's Application to the Learning of Inverted Pendulum System", 3rd IFSA Congress, Seattle, 1989, pp.610-613.

4) Takagi, Hideo and Hayashi, Isao, " Artificial Neural Network Driven Fuzzy Reasoning", Int. J. Approximate Reasoning, 1990, (in press).

5) Anderson, James and Rosenfeld, Edward, " Neurocomputing", MIT Press, Cambridge, Mass., 1988.

6) Tank, David and Hopplield, John, "Collective Computations in Neuronlike Circuits", Scientific American, December, 1987, pp. 104-114.
7) Rumlhart, David E., Hinton, Geoffrey E., and Williams, Ronald J., "Learning Representaions by Back-propagating Errors", Nature, Vol.323, No.9, 1986, pp.533-536.
8) Sugeno, Michio and Kang, G.T., "Structure identification of fuzzy model", Fuzzy Sets and Systems, Vol.28, No.1, 1988, pp.15-33.
9) Pao, Yoh-Han, "Adaptive Pattern Recognition and Neural Networks", Addison-Wesley, 1989.
10) Draper, N. R. and Smith, H., "Applied Regression Analysis", John Wiley & Sons, 1966.

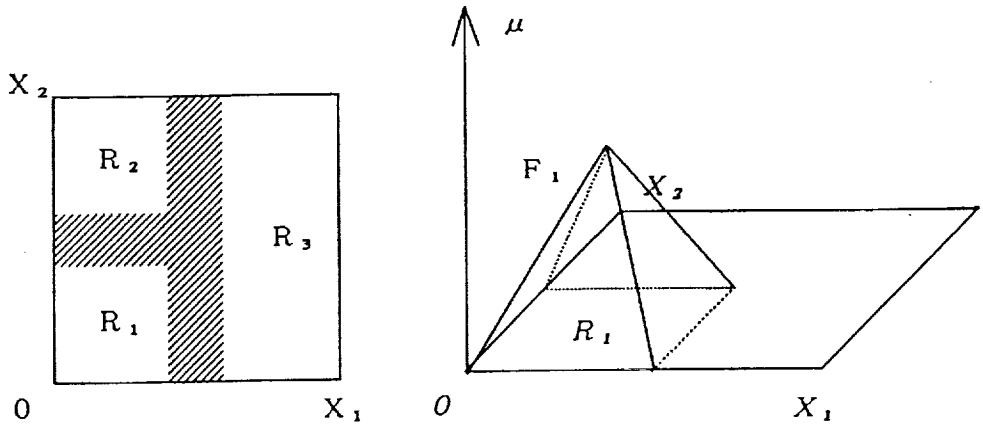

Fig.1 Conventional Fuzzy Partition of Rules

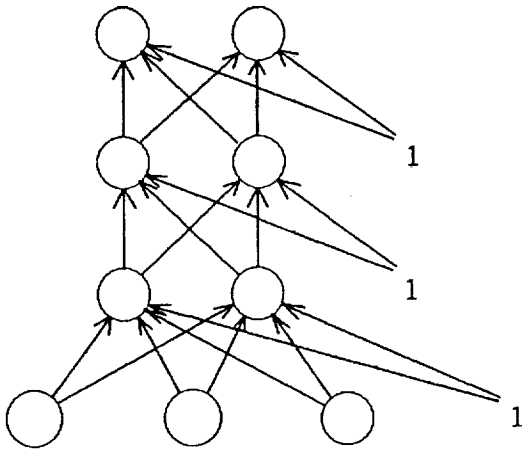

Fig.2 Example of Neural Network

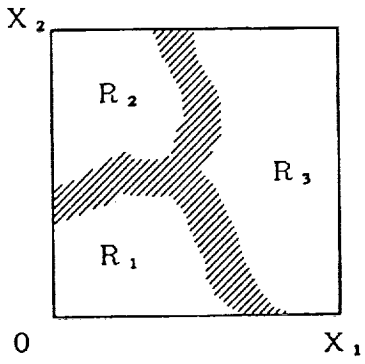

Fig.4 Proposed Fuzzy Partition of Rules

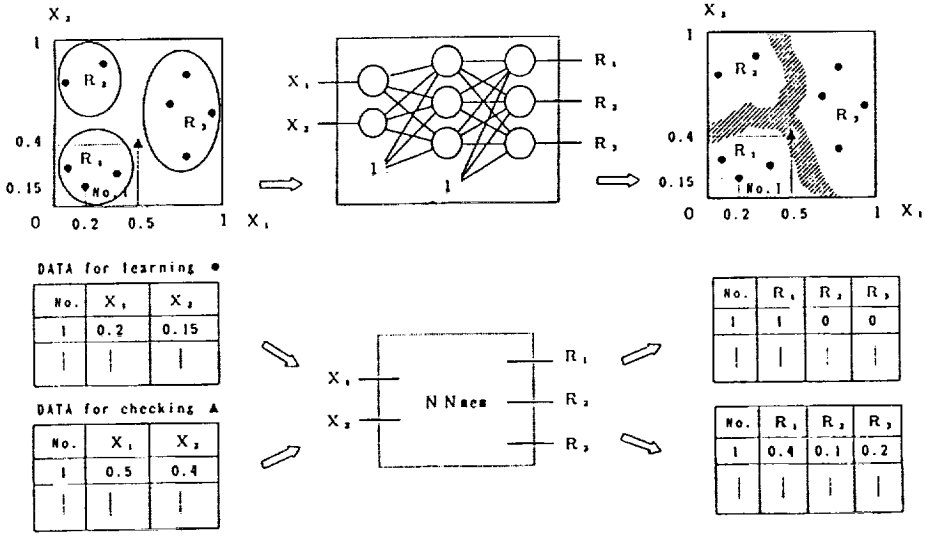

Fig.3 Decision of Membership Function in Antecedent Parts of Rules

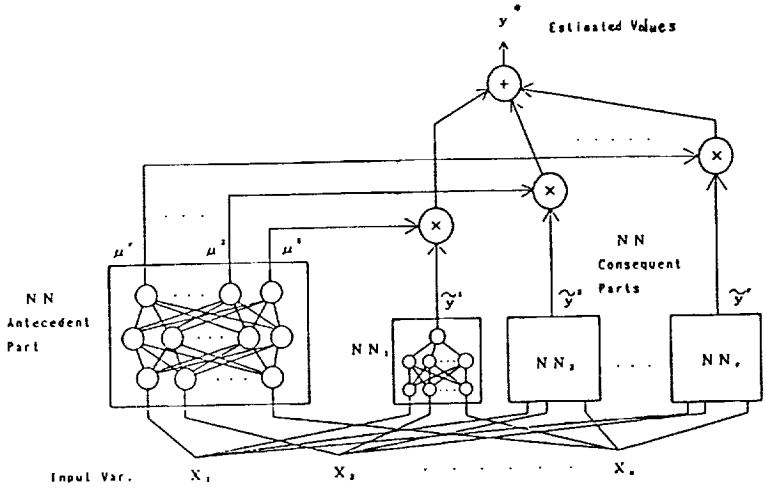

Fig.5 Block Diagram of Neural Network Driven Fuzzy Reasoning

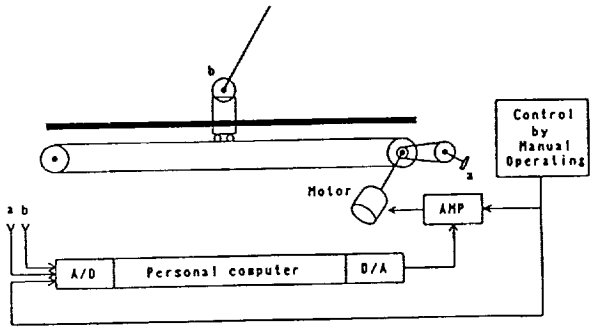

Fig.6 Structure of Inverted Pendulum System

Table1 Input and Output data of Inverted Pendulum System

No.	Input Data				Output Data
	x_1 [cm]	x_2 [cm/sec]	x_3 [deg]	x_4 [deg/sec]	y [v]
1	-1.1482	0.0000	178.5074	0.0000	0.7597
2	-0.0201	8.5486	180.9129	34.5660	0.7421
3	3.2197	29.9073	185.6439	34.5660	0.7617
4	7.8338	38.4472	188.4660	0.0000	0.0039
5	10.9510	4.2697	182.7386	-121.0536	-0.7168
6	9.3718	-21.3586	165.3085	-155.6554	-0.7968
7	5.3319	-38.4560	151.5283	-69.1678	-0.7519
8	0.1432	-42.7261	150.9487	51.8840	-0.7519
93	7.9980	42.7261	85.663	-380.4464	-0.0136
94	11.7713	4.2701	199.1743	-639.8375	-0.7265
95	10.6843	-17.0885	117.4961	-622.5536	-0.7519
96	7.0135	-42.7261	56.6514	-345.8786	-0.7519
97	1.9891	-38.4560	27.0170	-138.3357	0.0039
98	-2.9937	-34.1774	16.6419	-51.8822	-0.0019

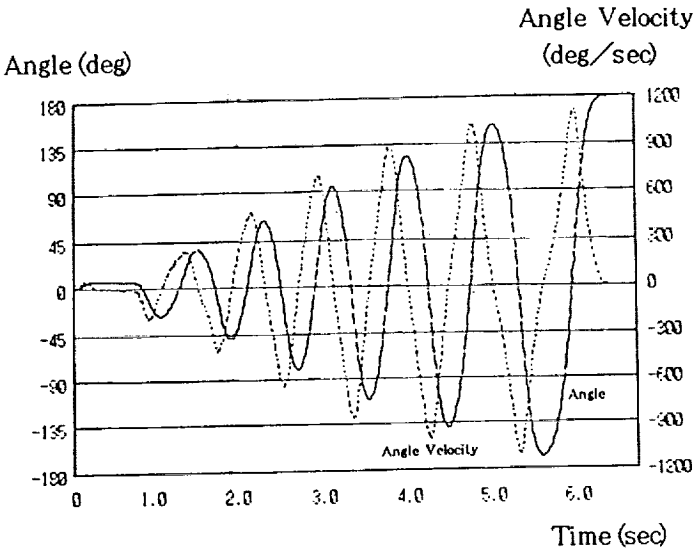

Fig.7 Angle and Angular Velocity of 40 cm Long Pendulum

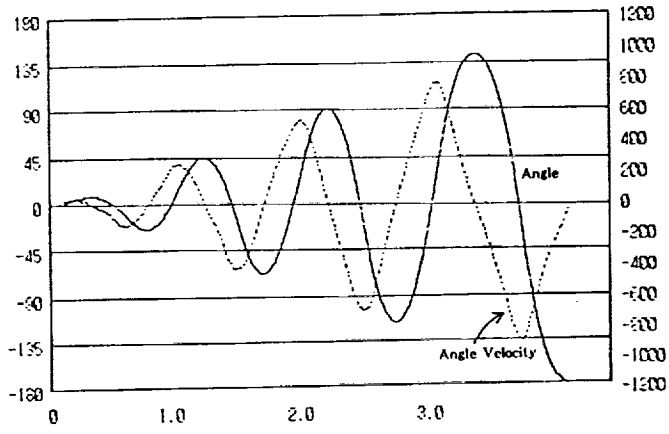

Fig.8 Angle and Angular Velocity of 20 cm Long Pendulum

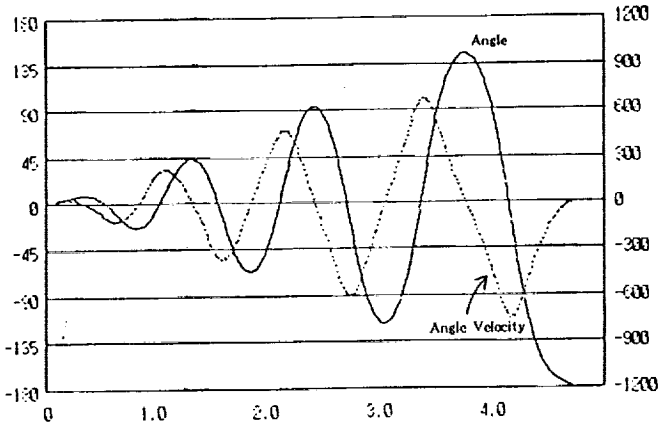

Fig.9 Angle and Angular Velocity of 30 cm Long Pendulum

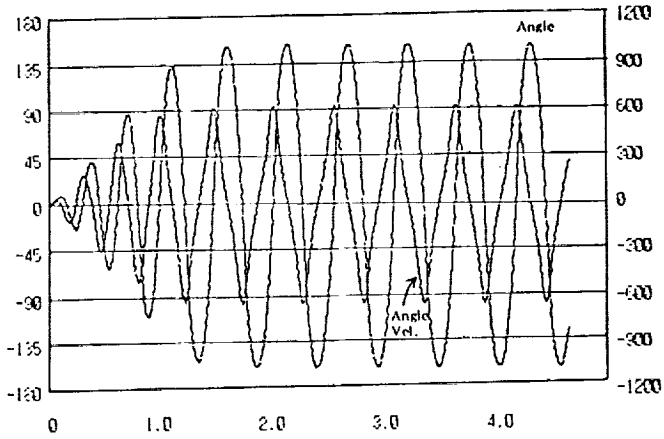

Fig.10 Angle and Angular Velocity of 50 cm Long Pendulum

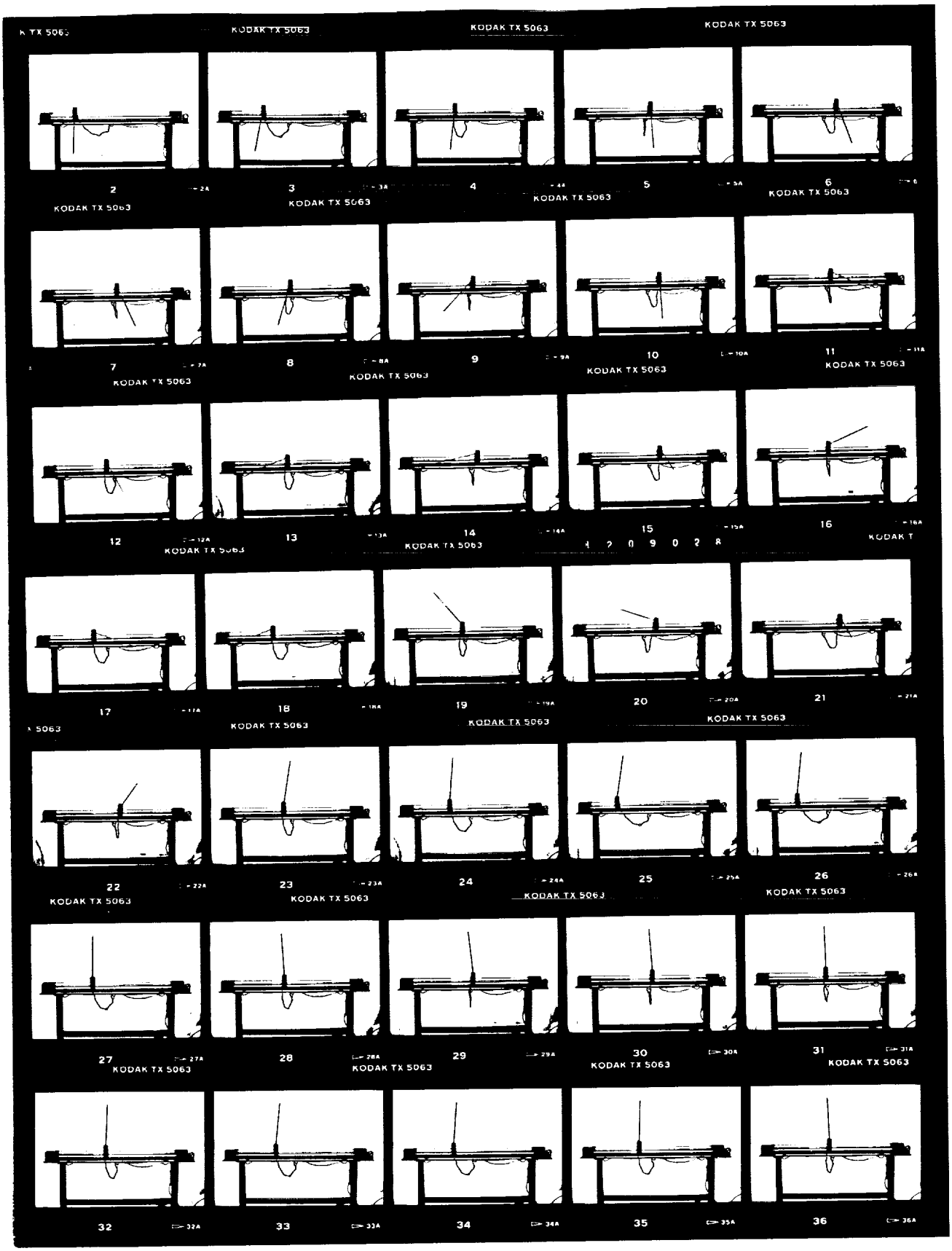

Photo.1 Control of Inverted Pendulum System (No.1)

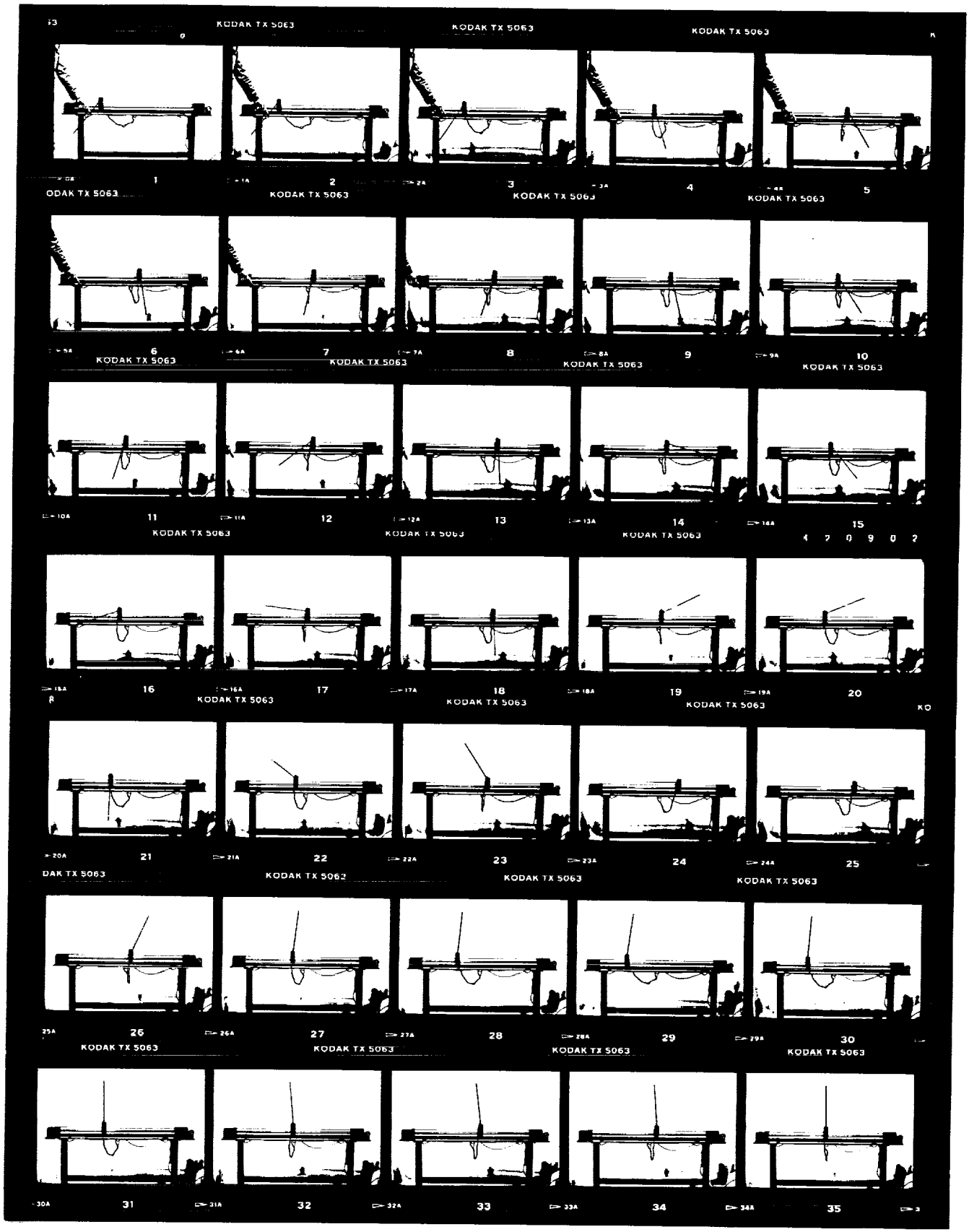

Photo.2 Control of Inverted Pendulum System (No.2)